CLAIMS

What is claimed is:

1   1.   A method comprising:

2        maintaining first and second return buffers for first and second instruction

3   pipelines respectively, both buffers having a plurality of return address entries; and

4        sending a request to the second instruction pipeline to fill the first return buffer

5   with the entries in the second return buffer.


1   2.   The method of claim 1, further comprising detecting a return instruction prior to

2   sending a request to the second instruction pipeline to fill the first return buffer with

3   the entries in the second return buffer.


1   3.   The method of claim 2, further comprising popping the first return buffer after

2   detecting a return instruction.


1   4.   The method of claim 2, further comprising determining if the return instruction

2   was detected by a third instruction pipeline and sending a return address in the first

3   return buffer to the third instruction pipeline if the return instruction was not detected

4   by the third instruction pipeline.


1   5.   The method of claim 3, further comprising updating a pointer to the first return

2   buffer after popping the first return buffer.


1   6.   The method of claim 2, further comprising detecting a call instruction prior to

2   detecting a return instruction.

1    7.     The method of claim 6, further comprising pushing a return address

2    corresponding to the call instruction onto the first return buffer after detecting a call

3    instruction.


1    8.     The method of claim 7, further comprising updating a pointer to the first return

2    buffer after pushing the return address onto the first return buffer.


1    9.     The method of claim 8, further comprising determining if the call instruction

2    was detected by a third instruction pipeline and sending a return address in the first

3    return buffer to the third instruction pipeline if the call instruction was not detected by

4    the third instruction pipeline.


1    10.    The method of claim 4, wherein determining if the return instruction was

2    detected by a third instruction pipeline comprises determining if the return instruction

3    was detected by a pipeline for predicting whether branches will be taken.


1    11.    The method of claim 1, wherein maintaining a first return buffer for a first

2    instruction pipeline comprises maintaining a first return buffer for a pipeline to

3    translate instructions fetched from a memory device.


1    12.    The method of claim 1, wherein maintaining a second return buffer for a

2    second instruction pipeline comprises maintaining a second return buffer for a

3    pipeline that predicts branches in a cache.


1    13.    The method of claim 7, wherein each entry in the first return buffer comprises a

2    valid bit that indicates whether the corresponding return address is valid.

1    14.    The method of claim 13, further comprising checking the valid bit

2    corresponding to each popped return address to determine if the return address is

3    valid.


1    15.    The method of claim 1, wherein the first and second return buffers are return

2    stacks.


1    16.    An instruction pipeline in a microprocessor comprising:

2          a decode unit, the decode unit to maintain a first return buffer having a plurality

3    of return address entries; and

4          a branch prediction unit, the branch prediction unit to maintain a second return

5    buffer and to fill the first return buffer with entries from the second return buffer.


1    17.    The instruction pipeline of claim 16, wherein the decode unit to further

2    recognize call instructions and return instructions.


1    18.    The instruction pipeline of claim 17, wherein the branch prediction unit to fill

2    the first return buffer with entries from the second return buffer comprises the branch

3    prediction unit to fill the first return buffer with entries from the second return buffer

4    each time the decode unit recognizes a return instruction.


1    19.    The instruction pipeline of claim 16, wherein decode unit to further send a

2    request to the branch prediction unit to fill the first return buffer with entries from the

3    second return buffer each time the first return buffer is not full.


1    20.    The instruction pipeline of claim 17, further comprising a second branch

2    prediction unit, the decode unit to send the second branch prediction unit a return

3    address in the first return buffer if the decode unit recognizes a call instruction or a

4    return instruction that the second branch prediction unit did not recognize.


1    21.    The instruction pipeline of claim 16, wherein the branch prediction unit

2    comprises a cache branch prediction unit to predict branches in a cache.


1    22.    The instruction pipeline of claim 21, wherein the cache branch prediction unit

2    to notify the decode unit when there is a cache miss and the decode unit to send a

3    request to the cache branch prediction unit to fill the first return buffer with entries

4    from the second return buffer.


1    23.    A method comprising:

2        maintaining a return buffer for a decode unit, the return buffer having a

3    plurality of return address entries;

4        detecting a call instruction;

5        determining if the call instruction was detected by a branch prediction unit;

6    and

7        sending a return address to the branch prediction unit if the call instruction was

8    not detected by the branch prediction unit.


1    24.    The method of claim 23, further comprising pushing a return address

2    corresponding to the call instruction onto the return buffer.


1    25.    The method of claim 23, further comprising detecting a return instruction.


1    26.    The method of claim 25, further comprising popping a return address from the

2    return stack buffer.

1   27.   The method of claim 26, wherein each entry in the return buffer comprises a

2   valid bit that indicates whether the return address is valid.


1   28.   The method of claim 27, further comprising checking the valid bit

2   corresponding to the popped return address to determine if the return address is

3   valid.


1   29.   The method of claim 23, further comprising sending a request to a cache

2   branch prediction unit to fill the return buffer.


1   30.   The method of claim 25, further comprising determining if the return instruction

2   was detected by the branch prediction unit and sending a return address in the return

3   buffer to the branch prediction unit if the return instruction was not detected by the

4   branch prediction unit.